Catgets

Be careful of function use of environment variables for search path

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-19

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 8101 bytes

Attack Categories	• Path spoofing or confusion problem • Malicious Input
Vulnerability Categories	• Format string • Buffer Overflow • Unconditional • Input source (not really attack)
Software Context	• National Language Support
Location	• nl_types.h
Description	Text obtained from message catalogs may not be trustworthy, and care must be exercised in how it is used. The function catopen() opens a message catalog file located either according to a supplied path (containing a "/" character) or by searching for a named catalog (with no "/") by referencing the values of the NLSPATH, LANG, and LC_MESSAGES environment variables. Subsequently, the catgets() function may be used to obtain message text from the catalog. If an attacker can influence the environment in which the program runs, he or she can cause a program to load strings from arbitrary files. Careless use of text returned by catgets() can create vulnerabilities that can be exploited by an attacker who manages to substitute text. Depending on how the text is used, buffer overflow or format vulnerabilities may be present, which, if exploited, could result in the execution of arbitrary code.
APIs	see below

FunctionName	Comments
catopen	opens message catalog based on environment
catgets	returns arbitrary length string

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

Method of Attack	Attacker can manipulate NLSPATH and related environment variables to control what gets returned by catopen() and catgets(). Alternatively, the standard catalog file could be overwritten if catalog directories are not secure. By installing a custom catalog of messages, the attacker can cause arbitrarily long strings to be returned and/or can include format string information (e.g., "%s") into the string, which may be interpreted if the text is used as a format string. In many cases, setuid programs access locale-specific message catalogs to print messages. If this is not done with due care, an attacker can use it to cause arbitrary code execution.
Exception Criteria	catgets() is safe if the standard catalog directory is secure and the catalog descriptor gotten from catopen() was opened using a fully specified path containing a "/" or NLSPATH and other environment variables are validated before being used. catgets() is also safe if the returned message text is used in a safe fashion.
Solutions	see table below

Solution Applicability	**Solution Description**	**Solution Efficacy**
Particularly applicable to setuid programs for which user can control environment.	Validate that catopen() will return an authentic message catalog. This requires that the message database (i.e., set of directories containing message catalog files) be in a secure, trusted directory. It also requires that either it be certain that an attacker could not manipulate the program environment (not necessarily an option for a setuid program) or that the information	Effective, but hard to implement correctly. Best used in combination with the solution of using text safely.

		used to locate the particular message catalog file be validated before catopen() is called. Specifying a fully qualified catalog path containing "/" would work but largely defeats the purpose of using a message catalog. The alternative is to examine NLSPATH and related environment variables to confirm that they correspond only to the expected secure directories. This also requires that the message catalog locations be constrained, with those constraints known to the program at compilation time.	
	Particularly applicable to setuid programs for which user can control environment.	Use text obtained from catgets() safely, in a way that reflects its untrustworthy nature. Text obtained from catgets() is typically used in printed or displayed messages. This should not be used as a format string, i.e., as in	Effective. Validating text to be used as a format string could be tricky unless rigid constraints are enforced.

	printf(text), but should instead be used as a data string, as in printf("%s", text). If the text must be used as a format string, then it should be parsed and validated as being safe before it is used. If text obtained from catgets() is placed in a buffer, care must be exercised to ensure that buffer overflows cannot occur.
Signature Details	Any use of catopen()/catgets() should be examined. If no checks are done on NLSPATH and usage of catgets() result matches signature for potential format string problem or buffer overflow problem, then a problem exists. Most relevant for setuid programs, for which user can control execution environment.

Examples of Incorrect Code	`nl_catd catd = catopen("MyCatalog", 0);`
	
	`char *text = catgets(catd, 2, 10, "Default text.");`
	`printf(text); // vulnerable to format string attack`
	`strcpy(buffer, text); // vulnerable to buffer overflow attack`

Examples of Corrected Code	`// Verify an expected secure path will be searched`
	`if (!nlsPathIsSafe())`
	`exit(EXIT_FAILURE);`
	`nl_catd catd = catopen("MyCatalog", 0);`
	
	`// Ensure safe usage of retrieved text`
	`char *text = catgets(catd, 2, 10, "Default text.");`

	`printf("%s", text);`
	`strncpy(buffer, text, bufferSize);`

Source References	• http://www.linuxsecurity.com/articles/hackscracks_article-1496.html
Recommended Resources	• catgets(3) man page[3] • catopen(3) man page[4]
Discriminant Set	**Operating Systems**: • UNIX (All) • Windows (All) **Languages**: • C • C++

Cigital, Inc. Copyright

1. mailto:copyright@cigital.com